

[illegible]

```
SSSSSSSS  UU      UU  BBBB BBBB  MM      MM  IIIIII  TTTTTTTTTT
SSSSSSSS  UU      UU  BBBB BBBB  MM      MM  IIIIII  TTTTTTTTTT
SS      SS  UU      UU  BB      BB  MMMM  MMMM  II      TT
SS      SS  UU      UU  BB      BB  MMMM  MMMM  II      TT
SS      SS  UU      UU  BB      BB  MM  MM  MM  II      TT
SSSSSS  SS  UU      UU  BBBB BBBB  MM      MM  II      TT
SSSSSS  SS  UU      UU  BBBB BBBB  MM      MM  II      TT
SS      SS  UU      UU  BB      BB  MM      MM  II      TT
SS      SS  UU      UU  BB      BB  MM      MM  II      TT
SS      SS  UU      UU  BB      BB  MM      MM  II      TT
SSSSSSSS  UUUUUUUUU  BBBB BBBB  MM      MM  IIIIII  TT
SSSSSSSS  UUUUUUUUU  BBBB BBBB  MM      MM  IIIIII  TT
```

```
LL      IIIIII  SSSSSSSS
LL      IIIIII  SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII  SSSSSSSS
LLLLLLLLLL  IIIIII  SSSSSSSS
```

```

0001 0 MODULE SUBMIT (XTITLE 'PRINT and SUBMIT Commands'
0002 0 MAIN = SUBMIT,
0003 0 IDENT = 'V04-000'
0004 0 ) =
0005 1 BEGIN
0006 1
0007 1 *****
0008 1 *
0009 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0010 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0011 1 * ALL RIGHTS RESERVED.
0012 1 *
0013 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0014 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0015 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0016 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0017 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0018 1 * TRANSFERRED.
0019 1 *
0020 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0021 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0022 1 * CORPORATION.
0023 1 *
0024 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0025 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0026 1 *
0027 1 *
0028 1 *****
0029 1
0030 1 ++
0031 1 FACILITY:
0032 1 PRINT and SUBMIT commands.
0033 1
0034 1 ABSTRACT:
0035 1 This module implements the PRINT and SUBMIT commands.
0036 1
0037 1 ENVIRONMENT:
0038 1 VAX/VMS user mode.
0039 1 --
0040 1
0041 1 AUTHOR: M. Jack, CREATION DATE: 30-Apr-1982
0042 1
0043 1 MODIFIED BY:
0044 1
0045 1 V03-009 RRB0021 Rowland R. Bradley 27-Aug-1984
0046 1 Disallow the submission of command procedures and
0047 1 and print jobs from sequential devices. Avoids the
0048 1 problem described in Crash 201 (symbiont in RFAST state).
0049 1 We do not support sharing sequential devices (mag tape).
0050 1 Clean up the error logging associated with PRINT and
0051 1 SUBMIT.
0052 1
0053 1 V03-008 JAK0220 J A Krycka 21-Jul-1984
0054 1 Make a $SNDJBC item list entry if /FEED is explicitly present.
0055 1 Before, an item list entry was made only for /NOFEED because
0056 1 /FEED was the default condition. Now the user can establish a
0057 1 queue level default value for this attribute, so it is

```


SUBMIT
V04-000

PRINT and SUBMIT Commands

K 10
16-Sep-1984 00:28:51
14-Sep-1984 12:09:55

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SUBMIT.B32;1

Page 2
(1)

```

58 0058 1  necessary to be able to override the queue default.
59 0059 1
60 0060 1  V03-007 RRB0020 Rowland R. Bradley      18-Jun-1984
61 0061 1  Add support for common command qualifier processing
62 0062 1  This is a late V4 enhancement ask for by customers
63 0063 1  to allow dumb user file filtering.
64 0064 1  Add enhanced error message handling for /REMOTE
65 0065 1  and remove redundant error message when no files
66 0066 1  in the job. Add REMOINSF error code.
67 0067 1
68 0068 1  V03-006 BLS0267      Benn Schreiber  31-Jan-1984
69 0069 1  Use new LIB$FIND_FILE argument.
70 0070 1
71 0071 1  V03-005 MLJ0115      Martin L. Jack, 29-Jul-1983  13:40
72 0072 1  Changes for job controller baselevel.
73 0073 1
74 0074 1  V03-004 MLJ0113      Martin L. Jack, 26-May-1983  17:00
75 0075 1  Changes for job controller baselevel.
76 0076 1
77 0077 1  V03-003 MLJ0112      Martin L. Jack, 29-Apr-1983  0:17
78 0078 1  Changes for job controller baselevel.
79 0079 1
80 0080 1  V03-002 MLJ0109      Martin L. Jack, 13-Apr-1983  14:44
81 0081 1  Disallow node name without /REMOTE.
82 0082 1
83 0083 1  V03-001 MLJ0106      Martin L. Jack, 1-Mar-1983   5:03
84 0084 1  Changes for job controller baselevel.
85 0085 1
86 0086 1  **

```

```
88 0087 1 LIBRARY 'SYSSLIBRARY:STARLET';
89 0088 1 REQUIRE 'SRC$:JBCPRSDEF';
90 0198 1
91 0199 1
92 0200 1 LITERAL
93 0201 1 TRUE= 1;
94 0202 1 FALSE= 0;
95 0203 1
96 0204 1
97 0205 1 STRUCTURE
98 0206 1 BBLOCK[O,P,S,E;N]=
99 0207 1 [N]
100 0208 1 (BBLOCK + 0)<P,S,E>;
101 0209 1
102 0210 1
103 0211 1 PSECT
104 0212 1 CODE= CODE,
105 0213 1 PLIT= CODE,
106 0214 1 OWN= DATA,
107 0215 1 GLOBAL= DATA;
108 0216 1
109 0217 1
110 0218 1 FORWARD ROUTINE
111 0219 1 SUBMIT,
112 0220 1 GET COMM QUAL_CONTEXT: NOVALUE,
113 0221 1 EXIT_HANDLER: NOVALUE,
114 0222 1 HANDLER;
115 0223 1
116 0224 1
117 0225 1 EXTERNAL ROUTINE
118 0226 1 CLISGET_VALUE: ADDRESSING_MODE(GENERAL),
119 0227 1 ! Get a parameter or qualifier value
120 0228 1 CLISPRESENT: ADDRESSING_MODE(GENERAL),
121 0229 1 ! Determine if entity is present
122 0230 1 LIB$FIND_FILE: ADDRESSING_MODE(GENERAL),
123 0231 1 ! Scan for wildcard files
124 0232 1 LIB$SCOPY_DXDX: ADDRESSING_MODE(GENERAL),
125 0233 1 ! Copy string
126 0234 1 LIB$QUAL_FILE_MATCH: ADDRESSING_MODE(GENERAL),
127 0235 1 ! Match common qualifiers
128 0236 1 LIB$QUAL_FILE_PARSE: ADDRESSING_MODE(GENERAL);
129 0237 1 ! Setup common qualifiers
130 0238 1
131 0239 1 EXTERNAL LITERAL
132 0240 1 ! Common Qualifier return status values.
133 0241 1
134 0242 1 LIB$QUIPRO, ! Quit Processing
135 0243 1 LIB$NEGANS, ! Negative Answer for this file
136 0244 1 LIB$FILFAIMAT, ! File failed to meet criteria
137 0245 1
138 0246 1 ! Parallel PRINT and SUBMIT messages.
139 0247 1
140 0248 1 PRINTS_FACILITY, SUBMITS_FACILITY,
141 0249 1 PRINTS_ADDFIL, SUBMITS_ADDFIL,
142 0250 1 PRINTS_CREJOB, SUBMITS_CREJOB,
143 0251 1 PRINTS_INVFILSPE, SUBMITS_INVFILSPE,
144 0252 1 PRINTS_REMOINSF, SUBMITS_REMOINSF,
```

SUBMIT
V04-000

PRINT and SUBMIT Commands

M 10
16-Sep-1984 00:28:51
14-Sep-1984 12:09:55

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SUBMIT.B32;1

Page 4
(2)

```
: 145      0253 1
: 146      0254 1      ! SUBMIT only messages.
: 147      0255 1
: 148      0256 1      SUBMIT$_INVLOGFIL;
: 149      0257 1
: 150      0258 1 GLOBAL
: 151      0259 1      QUAL_CONTEXT;          ! Context for Common Qualifiers
: 152      0260 1
: 153      0261 1 OWN
: 154      0262 1      EXIT_STATUS:  BBLOCK[4],    ! Most severe error status
: 155      0263 1      FLAGS:        BBLOCK[4];    ! Flag bits
: 156      0264 1
: 157      0265 1
: 158      0266 1 MACRO
: 159      0267 1      V_PRINT=      0,0,1,0 %;    ! PRINT command
: 160      0268 1      V_IDENTIFY=   0,1,1,0 %;    ! /IDENTIFY
: 161      0269 1      V_REMOTE=     0,2,1,0 %;    ! /REMOTE
: 162      0270 1      V_NO_LOG_FILE= 0,3,1,0 %;    ! /NOLOG_FILE
```



```
164 0271 1 ROUTINE SUBMIT=
165 0272 1
166 0273 1 ++
167 0274 1
168 0275 1 FUNCTIONAL DESCRIPTION:
169 0276 1 This routine is the main entry point for the PRINT and SUBMIT commands.
170 0277 1
171 0278 1 INPUT PARAMETERS:
172 0279 1 Standard VMS activation parameters (not used).
173 0280 1
174 0281 1 IMPLICIT INPUTS:
175 0282 1 NONE
176 0283 1
177 0284 1 OUTPUT PARAMETERS:
178 0285 1 NONE
179 0286 1
180 0287 1 IMPLICIT OUTPUTS:
181 0288 1 NONE
182 0289 1
183 0290 1 ROUTINE VALUE:
184 0291 1 Completion status.
185 0292 1
186 0293 1 SIDE EFFECTS:
187 0294 1 NONE
188 0295 1
189 0296 1 --
190 0297 1
191 0298 2 BEGIN
192 0299 2 LOCAL
193 0300 2 CNT: INITIAL(0), ! Number of files in job
194 0301 2 FIND_CONTEXT: REF BBLOCK, ! Context for LIB$FIND FILE
195 0302 2 FIND_RESULT: BBLOCK[DSC$C-S-BLN], ! Result of LIB$FIND FILE
196 0303 2 FIND_DEFAULT: BBLOCK[DSC$C-S-BLN], ! Default file for LIB$FIND FILE
197 0304 2 VALUE_DESC: BBLOCK[DSC$C-S-BLN], ! Qualifier value descriptor
198 0305 2 INPUT_DESC: BBLOCK[DSC$C-S-BLN], ! Input file specification
199 0306 2 LOG_FILE_DESC: BBLOCK[DSC$C-S-BLN], ! Log file descriptor
200 0307 2 JSM_BUFFER: VECTOR[132,BYTE], ! Job status message
201 0308 2 JSM_DESC: VECTOR[2], ! Descriptor for status message
202 0309 2 ITEM_BUFFER: BBLOCK[1024], ! $SNDJBC item buffer
203 0310 2 DATA_BUFFER: BBLOCK[1024], ! $SNDJBC data buffer
204 0311 2 EXIT_BLOCK: VECTOR[5], ! Exit control block
205 0312 2
206 0313 2 MACRO
207 0314 2 SD_[A] = BIND %NAME('D_', A) = $DESCRIPTOR(A) %;
208 0315 2 SD_(
209 0316 2 '$VERB',
210 0317 2 'P1',
211 0318 2 'AFTER',
212 0319 2 'BURST',
213 0320 2 'CHARACTERISTICS',
214 0321 2 'CLI',
215 0322 2 'COPIES',
216 0323 2 'CPU TIME',
217 0324 2 'DELETE',
218 0325 2 'DEVICE',
219 0326 2 'FEED',
220 0327 2 'FLAG',
220 0327 2 'FORM',
```

```
221 P 0328 2 'HEADER',
222 P 0329 2 'HOLD',
223 P 0330 2 'IDENTIFY',
224 P 0331 2 'JOB COUNT',
225 P 0332 2 'KEEP',
226 P 0333 2 'LOG FILE',
227 P 0334 2 'LOWERCASE',
228 P 0335 2 'NAME',
229 P 0336 2 'NOTE',
230 P 0337 2 'NOTIFY',
231 P 0338 2 'OPERATOR',
232 P 0339 2 'PAGES',
233 P 0340 2 'PARAMETERS',
234 P 0341 2 'PASSALL',
235 P 0342 2 'PRINTER',
236 P 0343 2 'PRIORITY',
237 P 0344 2 'QUEUE',
238 P 0345 2 'REMOTE',
239 P 0346 2 'RESTART',
240 P 0347 2 'SETUP',
241 P 0348 2 'SPACE',
242 P 0349 2 'TRAILER',
243 P 0350 2 'USER',
244 P 0351 2 'WSDEFAULT',
245 P 0352 2 'WSEXTENT',
246 P 0353 2 'WSQUOTA');
247 0354 BIND
248 0355 D_BURST_ALL= $DESCRIPTOR('BURST.ALL'),
249 0356 D_BURST_ONE= $DESCRIPTOR('BURST.ONE'),
250 0357 D_FLAG_ALL= $DESCRIPTOR('FLAG.ALL'),
251 0358 D_FLAG_ONE= $DESCRIPTOR('FLAG.ONE'),
252 0359 D_TRAILER_ALL= $DESCRIPTOR('TRAILER.ALL'),
253 0360 D_TRAILER_ONE= $DESCRIPTOR('TRAILER.ONE');
254 0361 BUILTIN
255 0362 FP;
256 0363
257 0364 PARSE_GLOBAL_REGISTERS;
258 0365
259 0366
260 0367 ! Initialize.
261 0368 !
262 0369 EXIT_STATUS = SS$_NORMAL;
263 0370 .FP = HANDLER;
264 0371
265 0372 INPUT_DESC[DSC$B_CLASS] = DSC$K_CLASS_D;
266 0373 INPUT_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
267 0374 INPUT_DESC[DSC$W_LENGTH] = 0;
268 0375 INPUT_DESC[DSC$A_POINTER] = 0;
269 0376
270 0377 VALUE_DESC[DSC$B_CLASS] = DSC$K_CLASS_D;
271 0378 VALUE_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
272 0379 VALUE_DESC[DSC$W_LENGTH] = 0;
273 0380 VALUE_DESC[DSC$A_POINTER] = 0;
274 0381
275 0382 LOG_FILE_DESC[DSC$B_CLASS] = DSC$K_CLASS_D;
276 0383 LOG_FILE_DESC[DSC$B_DTYPE] = DSC$K_DTYPE_T;
277 0384 LOG_FILE_DESC[DSC$W_LENGTH] = 0;
```



```
278 0385 2 LOG_FILE_DESC[DSC$A_POINTER] = 0;
279 0386 2
280 0387 2 FIND_RESULT[DSC$B_CLASS] = DSC$K_CLASS_D;
281 0388 2 FIND_RESULT[DSC$B_DTYPE] = DSC$K_DTYPE_T;
282 0389 2 FIND_RESULT[DSC$W_LENGTH] = 0;
283 0390 2 FIND_RESULT[DSC$A_POINTER] = 0;
284 0391 2
285 0392 2 FIND_CONTEXT = 0;
286 0393 2
287 0394 2 FIND_DEFAULT[DSC$B_CLASS] = DSC$K_CLASS_S;
288 0395 2 FIND_DEFAULT[DSC$B_DTYPE] = DSC$K_DTYPE_T;
289 0396 2 FIND_DEFAULT[DSC$W_LENGTH] = %CHARCOUNT('COM');
290 0397 2 FIND_DEFAULT[DSC$A_POINTER] = UPLIT BYTE('COM');
291 0398 2
292 0399 2 FLAGS = 0;
293 0400 2
294 0401 2
295 0402 2 ! Initialize for command parsing utilities.
296 0403 2
297 P 0404 2 PARSE_GLOBAL_INIT(
298 P 0405 2     ICURSOR= ITEM_BUFFER,
299 P 0406 2     DCURSOR= DATA_BUFFER,
300 P 0407 2     MESSAGE= SUBMIT$ FACILITY*16 + SHRS_INVQUAVAL + STS$K_SEVERE,
301 0408 2     VALUE_DESC= VALUE_DESC);
302 0409 2
303 0410 2
304 0411 2 ! Get the command verb. If it is PRINT, rather than SUBMIT, change some
305 0412 2 defaults.
306 0413 2
307 0414 2 CLISGET VALUE(D $VERB, VALUE_DESC);
308 0415 2 IF CH$RCHAR(.VALUE_DESC[DSC$A_POINTER]) EQL %C'P'
309 0416 2 THEN
310 0417 2     BEGIN
311 0418 2         FLAGS[V_PRINT] = TRUE;
312 0419 2         FIND_DEFAULT[DSC$A_POINTER] = UPLIT BYTE('LIS');
313 0420 2     END;
314 0421 2
315 0422 2 ! Get the common qualifiers context for the PRINT command
316 0423 2
317 0424 2 GET_COMM_QUAL_CONTEXT();
318 0425 2
319 0426 2
320 0427 2 ! Get the command qualifiers.
321 0428 2
322 0429 2 FLAGS[V_IDENTIFY] = CLISPRESENT(D_IDENTIFY);
323 0430 2 FLAGS[V_REMOTE] = CLISPRESENT(D_REMOTE);
324 0431 2
325 0432 2
326 0433 2 ! Get the /AFTER qualifier.
327 0434 2
328 0435 2 PARSE_CALL(AFTER, D_AFTER);
329 0436 2
330 0437 2
331 0438 2 ! Get the /CHARACTERISTICS qualifier.
332 0439 2
333 0440 2 PARSE_CALL(CHARACTERISTICS, D_CHARACTERISTICS);
334 0441 2
```

```
335 0442 2
336 0443 2 ! Get the /HOLD qualifier.
337 0444 2
338 P 0445 2 PARSE_CALL(IF_TRUE, D_HOLD,
339 0446 2 SJCS_HOLD);
340 0447 2
341 0448 2
342 0449 2 ! Get the /NOTIFY qualifier.
343 0450 2
344 P 0451 2 PARSE_CALL(IF_TRUE, D_NOTIFY,
345 0452 2 SJCS_NOTIFY);
346 0453 2
347 0454 2
348 0455 2 ! Get the /PARAMETERS qualifier.
349 0456 2
350 0457 2 PARSE_CALL(PARAMETERS, D_PARAMETERS);
351 0458 2
352 0459 2
353 0460 2 ! Get the /PRIORITY qualifier.
354 0461 2
355 0462 2 PARSE_CALL(PRIORITY, D_PRIORITY);
356 0463 2
357 0464 2
358 0465 2 ! Get the /RESTART qualifier.
359 0466 2
360 P 0467 2 PARSE_CALL(IF_TRUE FALSE, D_RESTART,
361 0468 2 SJCS_RESTART, SJCS_NO_RESTART);
362 0469 2
363 0470 2
364 0471 2 ! Get the /USER qualifier.
365 0472 2
366 0473 2 PARSE_CALL(USER, D_USER);
367 0474 2
368 0475 2
369 0476 2 IF .FLAGS[V_PRINT]
370 0477 2 THEN
371 0478 2 BEGIN
372 0479 2
373 0480 2 ! Get the command-level /BURST qualifier.
374 0481 2
375 0482 2 PARSE_CALL(IF_TRUE, D_BURST_ALL, SJCS_FILE_BURST);
376 0483 2 PARSE_CALL(IF_TRUE, D_BURST_ONE, SJCS_FILE_BURST_ONE);
377 0484 2 PARSE_CALL(IF_FALSE, D_BURST, SJCS_NO_FILE_BURST);
378 0485 2
379 0486 2
380 0487 2 ! Get the /DEVICE qualifier.
381 0488 2
382 0489 2 IF CLISPRESNT(D_DEVICE) AND NOT CLISPRESNT(D_QUEUE)
383 0490 2 THEN
384 0491 2 PARSE_CALL(QUEUE, D_DEVICE, SJCS_QUEUE, $DESCRIPTOR('SYSS$PRINT'));
385 0492 2
386 0493 2
387 0494 2 ! Get the command-level /FLAG qualifier.
388 0495 2
389 0496 2 PARSE_CALL(IF_TRUE, D_FLAG_ALL, SJCS_FILE_FLAG);
390 0497 2 PARSE_CALL(IF_TRUE, D_FLAG_ONE, SJCS_FILE_FLAG_ONE);
391 0498 2 PARSE_CALL(IF_FALSE, D_FLAG, SJCS_NO_FILE_FLAG);
```

```
392 0499
393 0500
394 0501      ! Get the /FORM qualifier.
395 0502      !
396 0503      PARSE_CALL(FORM, D_FORM);
397 0504
398 0505
399 0506      ! Get the /JOB_COUNT qualifier.
400 0507      !
401 P 0508      PARSE_CALL(COPIES, D_JOB_COUNT,
402 0509      SJCS_JOB_COPIES);
403 0510
404 0511
405 0512      ! Get the /LOWERCASE qualifier.
406 0513      !
407 P 0514      PARSE_CALL(IF TRUE, D_LOWERCASE,
408 0515      SJCS_LOWERCASE);
409 0516
410 0517
411 0518      ! Get the /NAME qualifier.
412 0519      !
413 0520      PARSE_CALL(NAME, D_NAME);
414 0521
415 0522
416 0523      ! Get the /NOTE qualifier.
417 0524      !
418 P 0525      PARSE_CALL(STRING, D_NOTE,
419 0526      SJCS_NOTE, SJCS_NO_NOTE);
420 0527
421 0528
422 0529      ! Get the /OPERATOR qualifier.
423 0530      !
424 P 0531      PARSE_CALL(STRING, D_OPERATOR,
425 0532      SJCS_OPERATOR_REQUEST, SJCS_NO_OPERATOR_REQUEST);
426 0533
427 0534
428 0535      ! Get the /QUEUE qualifier.
429 0536      !
430 0537      IF NOT CLISPRESNT(D_DEVICE)
431 0538      THEN
432 0539          PARSE_CALL(QUEUE, D_QUEUE, SJCS_QUEUE, $DESCRIPTOR('SYSS$PRINT'));
433 0540
434 0541
435 0542      ! Get the command-level /TRAILER qualifier.
436 0543      !
437 0544      PARSE_CALL(IF TRUE, D_TRAILER_ALL, SJCS_FILE_TRAILER);
438 0545      PARSE_CALL(IF TRUE, D_TRAILER_ONE, SJCS_FILE_TRAILER_ONE);
439 0546      PARSE_CALL(IF FALSE, D_TRAILER, SJCS_NO_FILE_TRAILER);
440 0547      END
441 0548      ELSE
442 0549      BEGIN
443 0550
444 0551      ! Get the /CLI qualifier.
445 0552      !
446 P 0553      PARSE_CALL(FILENAME, D_CLI,
447 0554      SJCS_CLI, SJCS_NO_CLI);
448 0555
```



```
449 0556
450 0557
451 0558
452 P 0559
453 0560
454 0561
455 0562
456 0563
457 0564
458 P 0565
459 0566
460 0567
461 0568
462 0569
463 0570
464 0571
465 0572
466 0573
467 0574
468 0575
469 0576
470 0577
471 0578
472 0579
473 0580
474 0581
475 0582
476 0583
477 0584
478 P 0585
479 P 0586
480 0587
481 0588
482 0589
483 0590
484 0591
485 0592
486 0593
487 0594
488 0595
489 0596
490 0597
491 0598
492 0599
493 0600
494 0601
495 P 0602
496 0603
497 0604
498 0605
499 0606
500 0607
501 P 0608
502 0609
503 0610
504 0611
505 0612
```

```
! Get the /CPUIME qualifier.
!
PARSE_CALL(CPUTIME, D_CPUTIME,
  SJCS_CPU_LIMIT, SJCS_NO_CPU_LIMIT);

! Get the /KEEP qualifier.
!
PARSE_CALL(IF TRUE, D_KEEP,
  SJCS_NO_LOG_DELETE);

! Get the /LOG_FILE qualifier.
!
Q_VALUE_DESC = LOG_FILE_DESC;
FLAGSV_NO_LOG_FILE] = PARSE_CALL_VALUE(LOG_FILE, D_LOG_FILE);

! Get the /NAME qualifier.
!
Q_VALUE_DESC = VALUE_DESC;
PARSE_CALL(NAME, D_NAME);

! Get the log file specification.
!
IF NOT .FLAGSV_NO_LOG_FILE]
THEN
  PARSE_CALL(NAME_AND_LOG_FILE,
    VALUE_DESC, LOG_FILE_DESC,
    $DESCRIPTOR('SYSSLOGIN:.LOG'), SUBMIT$_INVLOGFIL);

! Get the /PRINTER qualifier.
!
PARSE_CALL(PRINTER, D_PRINTER);

! Get the /QUEUE qualifier.
!
PARSE_CALL(QUEUE, D_QUEUE, SJCS_QUEUE, $DESCRIPTOR('SYSSBATCH'));

! Get the /WSDEFAULT qualifier.
!
PARSE_CALL(WORKING_SET, D_WSDEFAULT,
  SJCS_WSDEFAULT, SJCS_NO_WSDEFAULT);

! Get the /WSEXTENT qualifier.
!
PARSE_CALL(WORKING_SET, D_WSEXTENT,
  SJCS_WSEXTENT, SJCS_NO_WSEXTENT);

! Get the /WSQUOTA qualifier.
```

```
506      0613      !
507      P 0614      ! PARSE CALL(WORKING SET, D_WSQUOTA,
508      0615      !   SJCS_WSQUOTA, SJCS_NO_WSQUOTA);
509      0616      !
510      0617      !
511      0618      !
512      0619      ! Create the job.
513      0620      !
514      0621      ! IF NOT .FLAGS[V_REMOTE]
515      0622      ! THEN
516      0623      ! BEGIN
517      0624      !   LOCAL
518      0625      !     IOSB:          VECTOR[4,WORD], ! I/O status block
519      0626      !     STATUS:        ! Status return
520      0627      !
521      0628      !
522      0629      !   ! Finish the list and send the create job request.
523      0630      !
524      0631      !   Q_ICURSORE[0,0,32,0] = 0;
525      0632      !   STATUS = $SNDJBCW(
526      0633      !     FUNC=SJCS_CREATE_JOB,
527      0634      !     ITMLST=ITEM_BUFFER,
528      0635      !     IOSB=IOSB);
529      0636      !   IF .STATUS THEN STATUS = .IOSB;
530      0637      !   IF NOT .STATUS
531      0638      !   THEN
532      0639      !     SIGNAL STOP(
533      0640      !       SUBMITS_CREJOB, 0, .STATUS);
534      0641      !
535      0642      !
536      0643      !   ! Establish an exit handler to delete the open job on an abnormal exit.
537      0644      !
538      0645      !   EXIT_BLOCK[0] = 0;
539      0646      !   EXIT_BLOCK[1] = EXIT_HANDLER;
540      0647      !   EXIT_BLOCK[2] = 1;
541      0648      !   EXIT_BLOCK[3] = EXIT_BLOCK[4];
542      0649      !   EXIT_BLOCK[4] = 0;
543      0650      !   $DCLEXH(DESBLK=EXIT_BLOCK);
544      0651      ! END;
545      0652      !
546      0653      !
547      0654      ! Loop for all elements of the parameter list.
548      0655      !
549      0656      ! WHILE CLISGET_VALUE(D_P1, INPUT_DESC) DO
550      0657      ! BEGIN
551      0658      !   LOCAL
552      0659      !     STATUS_3,
553      0660      !     STATUS_2;
554      0661      !
555      0662      !   ! Set up the message buffer for the ADDFIL request.
556      0663      !
557      0664      !   Q_ICURSOR = ITEM_BUFFER;
558      0665      !   Q_DCURSOR = DATA_BUFFER;
559      0666      !
560      0667      !
561      0668      !   ! Get the /DELETE qualifier.
562      0669      !
```

```
563 P 0670 3 PARSE CALL(IF TRUE, D DELETE,  
564 0671 3 SJCS_DELETE_FILE);  
565 0672 3  
566 0673 3  
567 0674 3 IF .FLAGS[V_PRINT]  
568 0675 3 THEN  
569 0676 4 BEGIN  
570 0677 4  
571 0678 4 ! Get the local-level /BURST qualifier.  
572 0679 4  
573 P 0680 4 PARSE CALL(LOCAL TRUE FALSE, D BURST,  
574 0681 4 SJCS_FILE_BURST, SJCS_NO_FILE_BURST);  
575 0682 4  
576 0683 4  
577 0684 4 ! Get the /COPIES qualifier.  
578 0685 4  
579 P 0686 4 PARSE CALL(COPIES, D COPIES,  
580 0687 4 SJCS_FILE_COPIES);  
581 0688 4  
582 0689 4  
583 0690 4 ! Get the /FEED qualifier.  
584 0691 4  
585 P 0692 4 PARSE CALL(IF TRUE FALSE, D FEED,  
586 0693 4 SJCS_PAGINATE, SJCS_NO_PAGINATE);  
587 0694 4  
588 0695 4  
589 0696 4 ! Get the local-level /FLAG qualifier.  
590 0697 4  
591 P 0698 4 PARSE CALL(LOCAL TRUE FALSE, D FLAG,  
592 0699 4 SJCS_FILE_FLAG, SJCS_NO_FILE_FLAG);  
593 0700 4  
594 0701 4  
595 0702 4 ! Get the /HEADER qualifier.  
596 0703 4  
597 P 0704 4 PARSE CALL(IF TRUE, D HEADER,  
598 0705 4 SJCS_PAGE_HEADER);  
599 0706 4  
600 0707 4  
601 0708 4 ! Get the /PAGES qualifier.  
602 0709 4  
603 P 0710 4 PARSE CALL(LOWER UPPER, D PAGES,  
604 P 0711 4 SJCS_FIRST_PAGE, SJCS_NO_FIRST_PAGE,  
605 0712 4 SJCS_LAST_PAGE, SJCS_NO_LAST_PAGE);  
606 0713 4  
607 0714 4  
608 0715 4 ! Get the /PASSALL qualifier.  
609 0716 4  
610 P 0717 4 PARSE CALL(IF TRUE, D_PASSALL,  
611 0718 4 SJCS_PASSALL);  
612 0719 4  
613 0720 4  
614 0721 4 ! Get the /SETUP qualifier.  
615 0722 4  
616 P 0723 4 PARSE CALL(COMMA LIST, D SETUP,  
617 0724 4 SJCS_FILE_SETUP_MODULES, SJCS_NO_FILE_SETUP_MODULES);  
618 0725 4  
619 0726 4
```



```
620      0727 4      ! Get the /SPACE qualifier.
621      0728 4
622      P 0729 4      PARSE CALL(IF TRUE, D SPACE,
623      0730 4          SJCS_DOUBLE_SPACE);
624      0731 4
625      0732 4
626      0733 4      ! Get the local-level /TRAILER qualifier.
627      0734 4
628      P 0735 4      PARSE CALL(LOCAL TRUE FALSE, D TRAILER,
629      0736 4          SJCS_FILE_TRAILER, SJCS_NO_FILE_TRAILER);
630      0737 4      END;
631      0738 4
632      0739 4
633      0740 4      ! Loop, calling LIB$FIND_FILE to get files matching the input spec.
634      0741 4
635      0742 4      UNTIL
636      0743 4          BEGIN
637      0744 4              STATUS_2 = LIB$FIND_FILE(
638      0745 4                  INPUT_DESC,
639      0746 4                  FIND_RESULT,
640      0747 4                  FIND_CONTEXT,
641      0748 4                  FIND_DEFAULT,
642      0749 4                  0, 0, %REF(2));
643      0750 4              !Request multiple input file processing
644      0751 4              .STATUS_2 EQL RMSS_NMF
645      0752 4          END
646      0753 4      DO
647      0754 4          BEGIN
648      0755 4              IF NOT .STATUS_2
649      0756 4                  THEN
650      0757 4                      BEGIN
651      0758 4                          SIGNAL(
652      0759 4                              (IF .FLAGS[V PRINT]
653      0760 4                                  THEN SUBMIT$FACILITY*16 + SHRS_OPENIN + STSSK_ERROR
654      0761 4                                  ELSE SUBMIT$FACILITY*16 + SHRS_OPENIN + STSSK_SEVERE),
655      0762 4                                  1, FIND_RESULT,
656      0763 4                                  .FIND_CONTEXT[FAB$L_STS], .FIND_CONTEXT[FAB$L_STV]);
657      0764 4                      END
658      0765 4                  ELSE
659      0766 4                      BEGIN
660      0767 4                          LOCAL
661      0768 4                              PROMPT_DESC;
662      0769 4                          PROMPT_DESC = FIND_RESULT;
663      0770 4                              ! Pointer to a desc of
664      0771 4                              ! the filename
665      0772 4                          STATUS_3 = LIB$QUAL FILE_MATCH (
666      0773 4                              QUAL_CONTEXT,
667      0774 4                              FIND_CONTEXT,
668      0775 4                              0,
669      0776 4                              (' IF .FLAGS[V PRINT]
670      0777 4                                  THEN $DESCRIPTOR('!AS, Print? [N]:')
671      0778 4                                  ELSE $DESCRIPTOR('!AS, Submit? [N]:')),
672      0779 4                              PROMPT,
673      0780 4                              PROMPT_DESC,
674      0781 4                              0);
675      0782 4                              ! Prompt
676      0783 4                              ! Descriptor filename
677      0784 4                              ! No prompt routine
678      0785 4                          IF NOT (.STATUS_3)
```

```
677 0784 5 THEN
678 0785 6 BEGIN
679 0786 6 IF (.STATUS_3 NEQ LIB$ FILFAIMAT) AND
680 0787 6 (.STATUS_3 NEQ LIB$ QUIPRO) AND
681 0788 7 (.STATUS_3 NEQ LIB$ NEGANS)
682 0789 6 THEN
683 0790 7 BEGIN
684 0791 7 SIGNAL (SUBMIT$ FACILITY*16 +
685 0792 7 SHR$ FICNOTACC + STS$K_WARNING,
686 0793 7 1, FIND_RESULT);
687 0794 7 END
688 0795 7 ELSE IF (.STATUS_3 EQL LIB$ QUIPRO)
689 0796 6 THEN
690 0797 6 EXITLOOP; ! Quit Now
691 0798 6 END
692 0799 5 ELSE
693 0800 6 BEGIN
694 0801 6 FIND_CONTEXT[FAB$V_SPL] = FIND_CONTEXT[FAB$V_SCF] = FALSE;
695 0802 6
696 0803 6 IF NOT $OPEN(FAB=.FIND_CONTEXT)
697 0804 7 THEN
698 0805 6 SIGNAL(
699 0806 6 (IF .FLAGS[V_PRINT]
700 0807 7 THEN SUBMIT$ FACILITY*16 + SHR$ OPENIN + STS$K_ERROR
701 0808 7 ELSE SUBMIT$ FACILITY*16 + SHR$ OPENIN + STS$K_SEVERE),
702 0809 6 1, FIND_RESULT,
703 0810 6 .FIND_CONTEXT[FAB$L_STS], .FIND_CONTEXT[FAB$L_STV])
704 0811 6
705 0812 6 ELSE
706 0813 7 BEGIN
707 0814 7 IF .FLAGS[V_REMOTE]
708 0815 7 THEN
709 0816 8 BEGIN
710 0817 8 IF NOT .BBLOCK[FIND_CONTEXT[FAB$L_DEV], DEV$V_NET]
711 0818 8 THEN
712 0819 8 SIGNAL(
713 0820 8 SUBMIT$ REMOINSF,
714 0821 8 1, FIND_RESULT)
715 0822 8 ELSE
716 0823 8 IF .FLAGS[V_PRINT]
717 0824 8 THEN FIND_CONTEXT[FAB$V_SPL] = TRUE
718 0825 8 ELSE FIND_CONTEXT[FAB$V_SCF] = TRUE;
719 0826 8 END;
720 0827 7
721 0828 7 IF NOT $CLOSE(FAB=.FIND_CONTEXT)
722 0829 8 THEN
723 0830 7 SIGNAL(
724 0831 7 SUBMIT$ FACILITY*16 + SHR$ CLOSEIN + STS$K_ERROR,
725 0832 7 1, FIND_RESULT,
726 0833 7 .FIND_CONTEXT[FAB$L_STS], .FIND_CONTEXT[FAB$L_STV]);
727 0834 7
728 0835 7 IF NOT .FLAGS[V_REMOTE]
729 0836 7 THEN
730 0837 7 BEGIN
731 0838 7 IF .BBLOCK[FIND_CONTEXT[FAB$L_DEV], DEV$V_NET]
```

734	0841	8
735	0842	8
736	0843	8
737	0844	8
738	0845	8
739	0846	9
740	0847	9
741	0848	9
742	0849	9
743	0850	9
744	0851	9
745	0852	9
746	0853	9
747	0854	9
748	0855	10
749	0856	10
750	0857	10
751	0858	10
752	0859	10
753	0860	10
754	0861	10
755	0862	10
756	0863	10
757	0864	10
758	0865	10
759	0866	10
760	0867	10
761	0868	10
762	0869	10
763	0870	10
764	0871	10
765	0872	10
766	0873	10
767	0874	10
768	0875	10
769	0876	10
770	0877	10
771	0878	10
772	0879	10
773	0880	10
774	0881	10
775	0882	10
776	0883	10
777	0884	10
778	0885	10
779	0886	10
780	0887	9
781	0888	8
782	0889	7
783	0890	6
784	0891	5
785	0892	4
786	0893	3
787	0894	2
788	0895	2
789	0896	2
790	0897	2


```
791 0898 BEGIN
792 0899 LOCAL
793 0900     IOSB:          VECTOR[4,WORD], ! I/O status block
794 0901     STATUS:      ! Status return
795 0902
796 0903
797 0904 ! Add the JOB_STATUS_OUTPUT item and finish the list.
798 0905
799 0906 Q_ICURSOR = ITEM_BUFFER;
800 0907 IF .FLAGS[V_IDENTIFY]
801 0908 THEN
802 0909     BEGIN
803 0910         JSM_DESC[0] = 0;
804 0911         JSM_DESC[1] = JSM_BUFFER;
805 0912         Q_ICURSOR[0,0,16,0] = XALLOCATION(JSM_BUFFER);
806 0913         Q_ICURSOR[2,0,16,0] = SJCS_JOB_STATUS_OUTPUT;
807 0914         Q_ICURSOR[4,0,32,0] = JSM_BUFFER;
808 0915         Q_ICURSOR[8,0,32,0] = JSM_DESC;
809 0916         Q_ICURSOR = .Q_ICURSOR + T2;
810 0917     END;
811 0918 Q_ICURSOR[0,0,32,0] = 0;
812 0919
813 0920
814 0921 ! Send the close job request.
815 0922
816 0923 STATUS = $SNDJBCW(
817 0924     FUNC=SJCS_CLOSE_JOB,
818 0925     ITMLST=ITEM_BUFFER,
819 0926     IOSB=IOSB);
820 0927
821 0928
822 0929 ! Cancel the exit handler established to delete the job on abnormal exit.
823 0930
824 0931 $CANEXH(DESBLOCK=EXIT_BLOCK);
825 0932
826 0933
827 0934 ! If close job failed and the count of jobs sent indicates that there
828 0935 ! WERE jobs then issue error.
829 0936
830 0937 IF .STATUS THEN STATUS = .IOSB;
831 0938 IF NOT .STATUS AND (.CNT NEQ 0)
832 0939 THEN
833 0940     SIGNAL_STOP(
834 0941         SUBMIT$_CREJOB, 0, .STATUS);
835 0942
836 0943
837 0944 ! Execute /IDENTIFY only if there WAS a job. Assume an error
838 0945 ! was issued if no tasks in the job.
839 0946
840 0947 IF .FLAGS[V_IDENTIFY] AND (.CNT NEQ 0)
841 0948 THEN
842 0949     SIGNAL(
843 0950         SUBMIT$ FACILITY*16 + SHR$TEXT + ST$SK_SUCCESS,
844 0951         1 OR XB*0001*16,
845 0952         JSM_DESC);
846 0953
847 0954 END;
```

```
.. 848
.. 849
.. 850
.. 851
.. 852

0955 2
0956 2
0957 2
0958 2
0959 1
```

Return with the most severe error encountered.

EXIT_STATUS OR STSM_INHIB_MSG

END;

.TITLE SUBMIT PRINT and SUBMIT Commands
.IDENT \V04-000\

.PSECT DATA,NOEXE,2

00000 QUAL_CONTEXT::
 .BKLB 4
00004 EXIT_STATUS:
 .BKLB 4
00008 FLAGS: .BKLB 4

.PSECT CODE,NOWRT,2

```
42 52 45 56 24 00000 P.AAB: .ASCII \SVERB\
00005 .BKLB 3
00008 P.AAA: .LONG 5
0000C .ADDRESS P.AAB
31 50 00010 P.AAD: .ASCII \P1\
00012 .BKLB 2
00014 P.AAC: .LONG 2
00018 .ADDRESS P.AAD
52 45 54 46 41 0001C P.AAF: .ASCII \AFTER\
00021 .BKLB 3
00024 P.AAE: .LONG 5
00028 .ADDRESS P.AAF
54 53 52 55 42 0002C P.AAH: .ASCII \BURST\
00031 .BKLB 3
00034 P.AAG: .LONG 5
00038 .ADDRESS P.AAH
53 43 49 54 53 49 52 45 54 43 41 52 41 48 43 0003C P.AAJ: .ASCII \CHARACTERISTICS\
0004B .BKLB 1
0004C P.AAI: .LONG 15
00050 .ADDRESS P.AAJ
49 4C 43 00054 P.AAL: .ASCII \CLI\
00057 .BKLB 1
00058 P.AAK: .LONG 3
0005C .ADDRESS P.AAL
53 45 49 50 4F 43 00060 P.AAN: .ASCII \COPIES\
00066 .BKLB 2
00068 P.AAM: .LONG 6
0006C .ADDRESS P.AAN
45 4D 49 54 55 50 43 00070 P.AAP: .ASCII \CPU TIME\
00077 .BKLB 1
00078 P.AAQ: .LONG 7
0007C .ADDRESS P.AAP
45 54 45 4C 45 44 00080 P.AAR: .ASCII \DELETE\
00086 .BKLB 2
00088 P.AAQ: .LONG 6
0008C .ADDRESS P.AAR
45 43 49 56 45 44 00090 P.AAT: .ASCII \DEVICE\
```

PRINT and SUBMIT Commands

VAX-11 BLISS-32 V4.0-742
[CL IUTL.SRC]SUBMIT.B32;1

Page 18
(3)

Address	Offset	Value	Label	Comment
00000006	00096	P.AAS:	.BLKB	2
00000000	00098		.LONG	6
44 45 45 46	0009C	P.AAV:	.ADDRESS	P.AAT
00000004	000A0		.ASCII	\FEED\
00000000	000A4	P.AAU:	.LONG	4
00000000	000A8		.ADDRESS	P.AAV
47 41 4C 46	000AC	P.AAX:	.ASCII	\FLAG\
00000004	000B0	P.AAW:	.LONG	4
00000000	000B4		.ADDRESS	P.AAX
4D 52 4F 46	000B8	P.AAZ:	.ASCII	\FORM\
00000004	000BC	P.AAY:	.LONG	4
00000000	000C0		.ADDRESS	P.AAZ
52 45 44 41 45 48	000C4	P.ABB:	.ASCII	\HEADER\
	000CA		.BLKB	2
00000006	000CC	P.ABA:	.LONG	6
00000000	000D0		.ADDRESS	P.ABB
44 4C 4F 48	000D4	P.ABD:	.ASCII	\HOLD\
00000004	000D8	P.ABC:	.LONG	4
00000000	000DC		.ADDRESS	P.ABD
59 46 49 54 4E 45 44 49	000E0	P.ABF:	.ASCII	\IDENTIFY\
00000008	000E8	P.ABE:	.LONG	8
00000000	000EC		.ADDRESS	P.ABF
54 4E 55 4F 43 5F 42 4F 4A	000F0	P.ABH:	.ASCII	\JOB_COUNT\
	000F9		.BLKB	3
00000009	000FC	P.ABG:	.LONG	9
00000000	00100		.ADDRESS	P.ABH
50 45 45 4B	00104	P.ABJ:	.ASCII	\KEEP\
00000004	00108	P.ABI:	.LONG	4
00000000	0010C		.ADDRESS	P.ABJ
45 4C 49 46 5F 47 4F 4C	00110	P.ABL:	.ASCII	\LOG_FILE\
00000008	00118	P.ABK:	.LONG	8
00000000	0011C		.ADDRESS	P.ABL
45 53 41 43 52 45 57 4F 4C	00120	P.ABN:	.ASCII	\LOWERCASE\
	00129		.BLKB	3
00000009	0012C	P.ABN:	.LONG	9
00000000	00130		.ADDRESS	P.ABN
45 4D 41 4E	00134	P.ABP:	.ASCII	\NAME\
00000004	00138	P.ABO:	.LONG	4
00000000	0013C		.ADDRESS	P.ABP
45 54 4F 4E	00140	P.ABR:	.ASCII	\NOTE\
00000004	00144	P.ABQ:	.LONG	4
00000000	00148		.ADDRESS	P.ABR
59 46 49 54 4F 4E	0014C	P.ABT:	.ASCII	\NOTIFY\
	00152		.BLKB	2
00000006	00154	P.ABS:	.LONG	6
00000000	00158		.ADDRESS	P.ABT
52 4F 54 41 52 45 50 4F	0015C	P.ABV:	.ASCII	\OPERATOR\
00000008	00164	P.ABU:	.LONG	8
00000000	00168		.ADDRESS	P.ABV
53 45 47 41 50	0016C	P.ABX:	.ASCII	\PAGES\
	00171		.BLKB	3
00000005	00174	P.ABW:	.LONG	5
00000000	00178		.ADDRESS	P.ABX
53 52 45 54 45 4D 41 52 41 50	0017C	P.ABZ:	.ASCII	\PARAMETERS\
	00186		.BLKB	2
0000000A	00188	P.ABY:	.LONG	10
00000000	0018C		.ADDRESS	P.ABZ

4C	4C	41	53	53	41	50	00190	P.ACB:	.ASCII	\PASSALL\	:	
							00197		.BLKB	1	:	
						00000007	00198	P.ACA:	.LONG	7	:	
						00000000	0019C		.ADDRESS	P.ACB	:	
52	45	54	4E	49	52	50	001A0	P.ACD:	.ASCII	\PRINTER\	:	
							001A7		.BLKB	1	:	
						00000007	001A8	P.ACC:	.LONG	7	:	
						00000000	001AC		.ADDRESS	P.ACD	:	
59	54	49	52	4F	49	52	50	001B0	P.ACF:	.ASCII	\PRIORITY\	
						00000008	001B8	P.ACE:	.LONG	8	:	
						00000000	001BC		.ADDRESS	P.ACF	:	
		45	55	45	55	51	001C0	P.ACH:	.ASCII	\QUEUE\	:	
							001C5		.BLKB	3	:	
						00000005	001C8	P.ACG:	.LONG	5	:	
						00000000	001CC		.ADDRESS	P.ACH	:	
		45	54	4F	4D	45	52	001D0	P.ACJ:	.ASCII	\REMOTE\	
							001D6		.BLKB	2	:	
						00000006	001D8	P.ACI:	.LONG	6	:	
						00000000	001DC		.ADDRESS	P.ACJ	:	
54	52	41	54	53	45	52	001E0	P.ACL:	.ASCII	\RESTART\	:	
							001E7		.BLKB	1	:	
						00000007	001E8	P.ACK:	.LONG	7	:	
						00000000	001EC		.ADDRESS	P.ACL	:	
		50	55	54	45	53	001F0	P.ACN:	.ASCII	\SETUP\	:	
							001F5		.BLKB	3	:	
						00000005	001F8	P.ACM:	.LONG	5	:	
						00000000	001FC		.ADDRESS	P.ACN	:	
		45	43	41	50	53	00200	P.ACP:	.ASCII	\SPACE\	:	
							00205		.BLKB	3	:	
						00000005	00208	P.ACO:	.LONG	5	:	
						00000000	0020C		.ADDRESS	P.ACP	:	
52	45	4C	49	41	52	54	00210	P.ACR:	.ASCII	\TRAILER\	:	
							00217		.BLKB	1	:	
						00000007	00218	P.ACQ:	.LONG	7	:	
						00000000	0021C		.ADDRESS	P.ACR	:	
			52	45	53	55	00220	P.ACT:	.ASCII	\USER\	:	
						00000004	00224	P.ACS:	.LONG	4	:	
						00000000	00228		.ADDRESS	P.ACT	:	
54	4C	55	41	46	45	44	53	57	0022C	P.ACV:	.ASCII	\WSDEFAULT\
							00235		.BLKB	3	:	
						00000009	00238	P.ACU:	.LONG	9	:	
						00000000	0023C		.ADDRESS	P.ACV	:	
54	4E	45	54	58	45	53	57	00240	P.ACX:	.ASCII	\WSEXTENT\	
						00000008	00248	P.ACW:	.LONG	8	:	
						00000000	0024C		.ADDRESS	P.ACX	:	
		41	54	4F	55	51	53	57	00250	P.ACZ:	.ASCII	\WSQUOTA\
							00257		.BLKB	1	:	
						00000007	00258	P.ACY:	.LONG	7	:	
						00000000	0025C		.ADDRESS	P.ACZ	:	
4C	4C	41	2E	54	53	52	55	42	00260	P.ADB:	.ASCII	\BURST.ALL\
							00269		.BLKB	3	:	
						00000009	0026C	P.ADA:	.LONG	9	:	
						00000000	00270		.ADDRESS	P.ADB	:	
45	4E	4F	2E	54	53	52	55	42	00274	P.ADD:	.ASCII	\BURST.ONE\
							0027D		.BLKB	3	:	
						00000009	00280	P.ADC:	.LONG	9	:	
						00000000	00284		.ADDRESS	P.ADD	:	

```
      4C 4C 41 2E 47 41 4C 46 00288 P.ADF: .ASCII \FLAG.ALL\
      00000008 00290 P.ADE: .LONG 8
      00000000 00294 .ADDRESS P.ADF
      45 4E 4F 2E 47 41 4C 46 00298 P.ADH: .ASCII \FLAG.ONE\
      00000008 002A0 P.ADG: .LONG 8
      00000000 002A4 .ADDRESS P.ADH
      4C 4C 41 2E 52 45 4C 49 41 52 54 002A8 P.ADJ: .ASCII \TRAILER.ALL\
      00000008 002B3 .BLKB 1
      00000000 002B4 P.ADI: .LONG 11
      00000000 002B8 .ADDRESS P.ADJ
      45 4E 4F 2E 52 45 4C 49 41 52 54 002BC P.ADL: .ASCII \TRAILER.ONE\
      00000008 002C7 .BLKB 1
      00000000 002C8 P.ADK: .LONG 11
      00000000 002CC .ADDRESS P.ADL
      4D 4F 43 2E 002D0 P.ADM: .ASCII \.COM\
      53 49 4C 2E 002D4 P.ADN: .ASCII \.LIS\
      54 4E 49 52 50 24 53 59 53 002D8 P.ADP: .ASCII \SYSS$PRINT\
      00000009 002E1 .BLKB 3
      00000000 002E4 P.ADO: .LONG 9
      00000000 002E8 .ADDRESS P.ADP
      54 4E 49 52 50 24 53 59 53 002EC P.ADR: .ASCII \SYSS$PRINT\
      00000009 002F5 .BLKB 3
      00000000 002F8 P.ADQ: .LONG 9
      00000000 002FC .ADDRESS P.ADR
      47 4F 4C 2E 3A 4E 49 47 4F 4C 24 53 59 53 00300 P.ADT: .ASCII \SYSS$LOGIN:.LOG\
      0000000E 0030E .BLKB 2
      00000000 00310 P.ADS: .LONG 14
      00000000 00314 .ADDRESS P.ADT
      48 43 54 41 42 24 53 59 53 00318 P.ADV: .ASCII \SYSS$BATCH\
      00000009 00321 .BLKB 3
      00000000 00324 P.ADU: .LONG 9
      00000000 00328 .ADDRESS P.ADV
      5D 4E 5B 20 3F 74 6E 69 72 50 20 2C 53 41 21 0032C P.ADX: .ASCII \!AS, Print? [N]:\
      3A 0033B
      00000010 0033C P.ADW: .LONG 16
      00000000 00340 .ADDRESS P.ADX
      4E 5B 20 3F 74 69 6D 62 75 53 20 2C 53 41 21 00344 P.ADZ: .ASCII \!AS, Submit? [N]:\
      3A 5D 00353
      00000011 00355 .BLKB 3
      00000000 00358 P.ADY: .LONG 17
      00000000 0035C .ADDRESS P.ADZ
```

```
D-$VERB= P.AAA
D-P1= P.AAC
D-AFTER= P.AAE
D-BURST= P.AAG
D-CHARACTERISTICS= P.AAI
D-CLI= P.AAK
D-COPIES= P.AAM
D-CPUTIME= P.AAO
D-DELETE= P.AAQ
D-DEVICE= P.AAS
D-FEED= P.AAU
D-FLAG= P.AAW
D-FORM= P.AAY
D-HEADER= P.ABA
D-HOLD= P.ABC
```

D_IDENTIFY=	P.ABE
D_JOB_COUNT=	P.ABG
D_KEEP=	P.ABI
D_LOG_FILE=	P.ABK
D_LOWERCASE=	P.ABM
D_NAME=	P.ABO
D_NOTE=	P.ABQ
D_NOTIFY=	P.ABS
D_OPERATOR=	P.ABU
D_PAGES=	P.ABW
D_PARAMETERS=	P.ABY
D_PASSALL=	P.ACA
D_PRINTER=	P.ACC
D_PRIORITY=	P.ACE
D_QUEUE=	P.ACG
D_REMOTE=	P.ACI
D_RESTART=	P.ACK
D_SETUP=	P.ACM
D_SPACE=	P.ACO
D_TRAILER=	P.ACQ
D_USER=	P.ACS
D_WSDEFAULT=	P.ACU
D_WSEXTENT=	P.ACW
D_WSQUOTA=	P.ACY
D_BURST_ALL=	P.ADA
D_BURST_ONE=	P.ADC
D_FLAG_ALL=	P.ADE
D_FLAG_ONE=	P.ADG
D_TRAILER_ALL=	P.ADI
D_TRAILER_ONE=	P.ADK
.EXTRN	CLISGET VALUE, CLISPRESNT
.EXTRN	LIBSFIND_FILE, LIBSSCOPY_DXDX
.EXTRN	LIBSQUAL_FILE_MATCH
.EXTRN	LIBSQUAL_FILE_PARSE
.EXTRN	LIBS QUIPRO, LIBS NEGANS
.EXTRN	LIBS FILFAIMAT, PRINTS FACILITY
.EXTRN	SUBMITS FACILITY
.EXTRN	PRINTS_ADDFIL, SUBMITS_ADDFIL
.EXTRN	PRINTS_CREJOB, SUBMITS_CREJOB
.EXTRN	PRINTS_INVFILSPE
.EXTRN	SUBMITS_INVFILSPE
.EXTRN	PRINTS REMOINSF
.EXTRN	SUBMITS REMOINSF
.EXTRN	SUBMITS_INVLOGFIL
.EXTRN	PARSE_AFTER, PARSE CHARACTERISTICS
.EXTRN	PARSE_IF TRUE, PARSE PARAMETERS
.EXTRN	PARSE_PRIORITY, PARSE IF TRUE_FALSE
.EXTRN	PARSE_USER, PARSE IF FALSE
.EXTRN	PARSE_QUEUE, PARSE FORM
.EXTRN	PARSE_COPIES, PARSE_NAME
.EXTRN	PARSE_STRING, PARSE_FILENAME
.EXTRN	PARSE_CPUTIME, PARSE_LOG_FILE
.EXTRN	PARSE_NAME AND LOG FILE
.EXTRN	PARSE_PRINTER, PARSE WORKING_SET
.EXTRN	SYSSNDJBCW, SYSSDCLEXH
.EXTRN	PARSE_LOCAL TRUE FALSE
.EXTRN	PARSE_LOWER_UPPER

				.EXTRN	PARSE COMMA LIST		
				.EXTRN	SYSSOPEN, SYSSCLOSE		
				.EXTRN	SYSSCANEXH		
			OFFC 00000	SUBMIT:	.WORD	Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11	0271
	57	0000*	CF 9E 00002		MOVAB	FLAGS, R7	
	56	FD2D	CF 9E 00007		MOVAB	D_DEVICE, R6	
	5E	F724	CE 9E 0000C		MOVAB	-2268(SP), SP	
			54 D4 00011		CLRL	CNT	0298
	6D	0000V	CF 9E 00013		MOVAB	HANDLER, (FP)	0370
E0	AD	020E0000	8F D0 00018		MOVL	#34471936, INPUT_DESC	0374
		E4	AD D4 00020		CLRL	INPUT_DESC+4	0375
E8	AD	020E0000	8F D0 00023		MOVL	#3447T936, VALUE_DESC	0379
		EC	AD D4 0002B		CLRL	VALUE_DESC+4	0380
D8	AD	020E0000	8F D0 0002E		MOVL	#3447T936, LOG_FILE_DESC	0384
		DC	AD D4 00036		CLRL	LOG_FILE_DESC+4	0385
F8	AD	020E0000	8F D0 00039		MOVL	#34471936, FIND_RESULT	0389
		FC	AD D4 00041		CLRL	FIND_RESULT+4	0390
		04	AE D4 00044		CLRL	FIND_CONTEXT	0392
F0	AD	010E0004	8F D0 00047		MOVL	#17694724, FIND_DEFAULT	0396
F4	AD	0238	C6 9E 0004F		MOVAB	P.ADM, FIND_DEFAULT+4	0397
FC	A7		01 7D 00055		MOVQ	#1, EXIT STATUS	0369
	5B	0428	CE 9E 00059		MOVAB	ITEM_BUFFER, Q_ICURSOR	0408
	5A	28	AE 9E 0005E		MOVAB	DATA_BUFFER, Q_DCURSOR	
	59	00000000*	8F D0 00062		MOVL	#<<<SUBMIT\$_FACILITY\$16>+4904>+4>, -	
						Q MESSAGE	
	58	E8	AD 9E 00069		MOVAB	VALUE_DESC, Q_VALUE_DESC	
		E8	AD 9F 0006D		PUSHAB	VALUE_DESC	0414
		FF70	C6 9F 00070		PUSHAB	D \$VERB	
00000000G	00		02 FB 00074		CALLS	#2, CLISGET VALUE	
50	8F	EC	BD 91 0007B		CMPI	@VALUE_DESC+4, #80	0415
			09 12 00080		BNEQ	1\$	
	67		01 88 00082		BISB2	#1, FLAGS	0418
F4	AD	023C	C6 9E 00085		MOVAB	P.ADM, FIND_DEFAULT+4	0419
0000V	CF		00 FB 0008B	1\$:	CALLS	#0, GET COMM_QUAL_CONTEXT	0424
		50	A6 9F 00090		PUSHAB	D IDENTIFY	0429
00000000G	00		01 FB 00093		CALLS	#T, CLISPRESENT	
01	01		50 F0 0009A		INSV	RO, #1, #1, FLAGS	
		0140	C6 9F 0009F		PUSHAB	D REMOTE	0430
00000000G	00		01 FB 000A3		CALLS	#T, CLISPRESENT	
01	02		50 F0 000AA		INSV	RO, #2, #1, FLAGS	
		8C	A6 9F 000AF		PUSHAB	D AFTER	0435
0000G	CF		01 FB 000B2		CALLS	#T, PARSE AFTER	
		84	A6 9F 000B7		PUSHAB	D CHARACTERISTICS	0440
0000G	CF		01 FB 000BA		CALLS	#T, PARSE CHARACTERISTICS	
	7E	47	8F 9A 000BF		MOVZBL	#71, -(SP)	0446
		40	A6 9F 000C3		PUSHAB	D HOLD	
0000G	CF		02 FB 000C6		CALLS	#2, PARSE IF_TRUE	
	7E	6C	8F 9A 000CB		MOVZBL	#108, -(SP)	0452
		00BC	C6 9F 000CF		PUSHAB	D NOTIFY	
0000G	CF		02 FB 000D3		CALLS	#2, PARSE IF_TRUE	
		00F0	C6 9F 000D8		PUSHAB	D PARAMETERS	0457
0000G	CF		01 FB 000DC		CALLS	#T, PARSE PARAMETERS	
		0120	C6 9F 000E1		PUSHAB	D PRIORITY	0462
0000G	CF		01 FB 000E5		CALLS	#T, PARSE PRIORITY	
	7E	8B	8F 9A 000EA		MOVZBL	#139, -(SP)	0468
	7E	8A	8F 9A 000EE		MOVZBL	#138, -(SP)	

		0150	C6	9F	000F2	PUSHAB	D RESTART		
0000G	CF		03	FB	000F6	CALLS	#3, PARSE_IF_TRUE_FALSE		
		018C	C6	9F	000FB	PUSHAB	D USER		0473
0000G	CF		01	FB	000FF	CALLS	#T, PARSE_USER		
	03		67	FB	00104	BLBS	FLAGS, 2\$		0476
			00F6	31	00107	BRW	5\$		
			20	DD	0010A	PUSHL	#32		0482
		01D4	C6	9F	0010C	PUSHAB	D BURST ALL		
0000G	CF		02	FB	00110	CALLS	#2, PARSE_IF_TRUE		
			21	DD	00115	PUSHL	#33		0483
		01E8	C6	9F	00117	PUSHAB	D BURST ONE		
0000G	CF		02	FB	0011B	CALLS	#2, PARSE_IF_TRUE		
			22	DD	00120	PUSHL	#34		0484
		9C	A6	9F	00122	PUSHAB	D BURST		
0000G	CF		02	FB	00125	CALLS	#2, PARSE_IF_FALSE		
			56	DD	0012A	PUSHL	R6		0489
00000000G	00		01	FB	0012C	CALLS	#1, CLISPRESNT		
	1D		50	E9	00133	BLBC	R0, 3\$		
		0130	C6	9F	00136	PUSHAB	D QUEUE		
00000000G	00		01	FB	0013A	CALLS	#T, CLISPRESNT		
	0F		50	E8	00141	BLBS	R0, 3\$		
		024C	C6	9F	00144	PUSHAB	P.ADD		0491
	7E	86	8F	9A	00148	MOVZBL	#134, -(SP)		
			56	DD	0014C	PUSHL	R6		
0000G	CF		03	FB	0014E	CALLS	#3, PARSE_QUEUE		
			24	DD	00153	PUSHL	#36		0496
		01F8	C6	9F	00155	PUSHAB	D FLAG ALL		
0000G	CF		02	FB	00159	CALLS	#2, PARSE_IF_TRUE		
			25	DD	0015E	PUSHL	#37		0497
		0208	C6	9F	00160	PUSHAB	D FLAG ONE		
0000G	CF		02	FB	00164	CALLS	#2, PARSE_IF_TRUE		
			26	DD	00169	PUSHL	#38		0498
		18	A6	9F	0016B	PUSHAB	D FLAG		
0000G	CF		02	FB	0016E	CALLS	#2, PARSE_IF_FALSE		
		24	A6	9F	00173	PUSHAB	D FORM		0503
0000G	CF		01	FB	00176	CALLS	#T, PARSE_FORM		
	7E	4B	8F	9A	0017B	MOVZBL	#75, -(SP)		0509
		64	A6	9F	0017F	PUSHAB	D JOB COUNT		
0000G	CF		02	FB	00182	CALLS	#2, PARSE_COPIES		
	7E	66	8F	9A	00187	MOVZBL	#102, -(SP)		0515
		0094	C6	9F	0018B	PUSHAB	D LOWERCASE		
0000G	CF		02	FB	0018F	CALLS	#2, PARSE_IF_TRUE		
		00A0	C6	9F	00194	PUSHAB	D NAME		0520
0000G	CF		01	FB	00198	CALLS	#T, PARSE_NAME		
	7E	6B	8F	9A	0019D	MOVZBL	#107, -(SP)		0526
	7E	6A	8F	9A	001A1	MOVZBL	#106, -(SP)		
		00AC	C6	9F	001A5	PUSHAB	D NOTE		
0000G	CF		03	FB	001A9	CALLS	#3, PARSE_STRING		
	7E	6F	8F	9A	001AE	MOVZBL	#111, -(SP)		0532
	7E	6E	8F	9A	001B2	MOVZBL	#110, -(SP)		
		00CC	C6	9F	001B6	PUSHAB	D OPERATOR		
0000G	CF		03	FB	001BA	CALLS	#3, PARSE_STRING		
			56	DD	001BF	PUSHL	R6		0537
00000000G	00		01	FB	001C1	CALLS	#1, CLISPRESNT		
	11		50	E8	001CB	BLBS	R0, 4\$		
		0260	C6	9F	001CB	PUSHAB	P.ADD		0539
	7E	86	8F	9A	001CF	MOVZBL	#134, -(SP)		

67

01

15

51

0000G	CF	0130	C6	9F	001D3	PUSHAB	D QUEUE	
			03	FB	001D7	CALLS	#3, PARSE_QUEUE	
			2B	DD	001DC	PUSHL	#43	0544
		021C	C6	9F	001DE	PUSHAB	D TRAILER_ALL	
0000G	CF		02	FB	001E2	CALLS	#2, PARSE_IF_TRUE	
			2C	DD	001E7	PUSHL	#44	0545
		0230	C6	9F	001E9	PUSHAB	D TRAILER_ONE	
0000G	CF		02	FB	001ED	CALLS	#2, PARSE_IF_TRUE	
			2D	DD	001F2	PUSHL	#45	0546
		0180	C6	9F	001F4	PUSHAB	D TRAILER	
0000G	CF		02	FB	001F8	CALLS	#2, PARSE_IF_FALSE	
			00A9	31	001FD	BRW	78	0476
			12	DD	00200	PUSHL	#18	0554
			11	DD	00202	PUSHL	#17	
		C0	A6	9F	00204	PUSHAB	D CLI	
0000G	CF		03	FB	00207	CALLS	#3, PARSE_FILENAME	
			16	DD	0020C	PUSHL	#22	0560
			15	DD	0020E	PUSHL	#21	
		E0	A6	9F	00210	PUSHAB	D CPUTIME	
0000G	CF		03	FB	00213	CALLS	#3, PARSE_CPUTIME	
	7E	60	8F	9A	00218	MOVZBL	#96, -(SP)	0566
		70	A6	9F	0021C	PUSHAB	D KEEP	
0000G	CF		02	FB	0021F	CALLS	#2, PARSE IF TRUE	
	58	D8	AD	9E	00224	MOVAB	LOG FILE DESC, Q_VALUE_DESC	0571
		0080	C6	9F	00228	PUSHAB	D LOG FILE	0572
0000G	CF		01	FB	0022C	CALLS	#T, PARSE_LOG_FILE	
	03		50	FO	00231	INSV	RO, #3, #T, FLAGS	
	58	E8	AD	9E	00236	MOVAB	VALUE DESC, Q_VALUE_DESC	0577
		00A0	C6	9F	0023A	PUSHAB	D NAME	0578
0000G	CF		01	FB	0023E	CALLS	#T, PARSE_NAME	
	67		03	E0	00243	BBS	#3, FLAGS, 68	0583
		00000000G	8F	DD	00247	PUSHL	#SUBMITS_INVLOGFIL	0587
		0278	C6	9F	0024D	PUSHAB	P.ADS	
		D8	AD	9F	00251	PUSHAB	LOG FILE DESC	
		E8	AD	9F	00254	PUSHAB	VALUE DESC	
0000G	CF		04	FB	00257	CALLS	#4, PARSE_NAME_AND_LOG_FILE	
		0110	C6	9F	0025C	PUSHAB	D PRINTER	0592
0000G	CF		01	FB	00260	CALLS	#T, PARSE_PRINTER	
		028C	C6	9F	00265	PUSHAB	P.ADU	0597
	7E	86	8F	9A	00269	MOVZBL	#134, -(SP)	
		0130	C6	9F	0026D	PUSHAB	D QUEUE	
0000G	CF		03	FB	00271	CALLS	#3, PARSE_QUEUE	
	7E	98	8F	9A	00276	MOVZBL	#152, -(SP)	0603
	7E	97	8F	9A	0027A	MOVZBL	#151, -(SP)	
		01A0	C6	9F	0027E	PUSHAB	D W\$DEFAULT	
0000G	CF		03	FB	00282	CALLS	#3, PARSE_WORKING_SET	
	7E	9A	8F	9A	00287	MOVZBL	#154, -(SP)	0609
	7E	99	8F	9A	0028B	MOVZBL	#153, -(SP)	
		0180	C6	9F	0028F	PUSHAB	D W\$EXTENT	
0000G	CF		03	FB	00293	CALLS	#3, PARSE_WORKING_SET	
	7E	9C	8F	9A	00298	MOVZBL	#156, -(SP)	0615
	7E	9B	8F	9A	0029C	MOVZBL	#155, -(SP)	
		01C0	C6	9F	002A0	PUSHAB	D W\$QUOTA	
0000G	CF		03	FB	002A4	CALLS	#3, PARSE_WORKING_SET	
	67		02	E0	002A9	BBS	#2, FLAGS, 108	0621
			6B	D4	002AD	CLRL	(Q ICURSOR)	0631
			7E	7C	002AF	CLRQ	-(SP)	0635

		14	AE	9F	002B1	PUSHAB	IOSB		
		0434	CE	9F	002B4	PUSHAB	ITEM_BUFFER		
	7E		0A	7D	002B8	MOVQ	#10, -(SP)		
			7E	D4	002BB	CLRL	-(SP)		
00000000G	00		07	FB	002BD	CALLS	#7, SYSSNDJBCW		
	07		50	E9	002C4	BLBC	STATUS, 8\$		0636
	50	0C	AE	D0	002C7	MOVL	IOSB, STATUS		
	11		50	E8	002CB	BLBS	STATUS, 9\$		0637
			50	DD	002CE	PUSHL	STATUS		0640
			7E	D4	002D0	CLRL	-(SP)		0639
		00000000G	8F	DD	002D2	PUSHL	#SUBMITS CREJOB		
00000000G	00		03	FB	002D8	CALLS	#3, LIB\$STOP		
		14	AE	D4	002DF	CLRL	EXIT_BLOCK		0645
	18	0000V	CF	9E	002E2	MOVAB	EXIT_HANDLER, EXIT_BLOCK+4		0646
	1C		01	D0	002E8	MOVL	#1, EXIT_BLOCK+8		0647
	20		AE	9E	002EC	MOVAB	EXIT_BLOCK+16, EXIT_BLOCK+12		0648
		24	AE	D4	002F1	CLRL	EXIT_BLOCK+16		0649
		14	AE	9F	002F4	PUSHAB	EXIT_BLOCK		0650
00000000G	00		01	FB	002F7	CALLS	#1, SYSSDCLEXH		
		EO	AD	9F	002FE	PUSHAB	INPUT_DESC		0656
		FF7C	C6	9F	00301	PUSHAB	D P1		
00000000G	00		02	FB	00305	CALLS	#2, CLISGET_VALUE		
	03		50	E8	0030C	BLBS	R0, 11\$		
		0236	31	0030F	BRW	35\$			
	5B	0428	CE	9E	00312	MOVAB	ITEM_BUFFER, Q_ICURSOR		0664
	5A	28	AE	9E	00317	MOVAB	DATA_BUFFER, Q_DCURSOR		0665
			18	DD	0031B	PUSHL	#24		0671
		FO	A6	9F	0031D	PUSHAB	D DELETE		
0000G	CF		02	FB	00320	CALLS	#2, PARSE_IF_TRUE		
	03		67	E8	00325	BLBS	FLAGS, 12\$		0674
		0085	31	00328	BRW	13\$			
			22	DD	0032B	PUSHL	#34		0681
			20	DD	0032D	PUSHL	#32		
		9C	A6	9F	0032F	PUSHAB	D BURST		
0000G	CF		03	FB	00332	CALLS	#3, PARSE_LOCAL_TRUE_FALSE		
			23	DD	00337	PUSHL	#35		0687
		D0	A6	9F	00339	PUSHAB	D COPIES		
0000G	CF		02	FB	0033C	CALLS	#2, PARSE COPIES		
	7E	76	8F	9A	00341	MOVZBL	#118, -(SP)		0693
	7E	75	8F	9A	00345	MOVZBL	#117, -(SP)		
		0C	A6	9F	00349	PUSHAB	D FEED		
0000G	CF		03	FB	0034C	CALLS	#3, PARSE_IF_TRUE_FALSE		
			26	DD	00351	PUSHL	#38		0699
			24	DD	00353	PUSHL	#36		
		18	A6	9F	00355	PUSHAB	D FLAG		
0000G	CF		03	FB	00358	CALLS	#3, PARSE_LOCAL_TRUE_FALSE		
	7E	71	8F	9A	0035D	MOVZBL	#113, -(SP)		0705
		34	A6	9F	00361	PUSHAB	D HEADER		
0000G	CF		02	FB	00364	CALLS	#2, PARSE_IF_TRUE		
	7E	5C	8F	9A	00369	MOVZBL	#92, -(SP)		0712
	7E	5B	8F	9A	0036D	MOVZBL	#91, -(SP)		
			2F	DD	00371	PUSHL	#47		
			2E	DD	00373	PUSHL	#46		
		00DC	C6	9F	00375	PUSHAB	D PAGES		
0000G	CF		05	FB	00379	CALLS	#5, PARSE_LOWER_UPPER		
	7E	80	8F	9A	0037E	MOVZBL	#128, -(SP)		0718
		0100	C6	9F	00382	PUSHAB	D_PASSALL		

0000G	CF		02	FB	00386	CALLS	#2, PARSE_IF_TRUE		
			29	DD	00388	PUSHL	#41		0724
			28	DD	0038D	PUSHL	#40		
		0160	C6	9F	0038F	PUSHAB	D SETUP		
0000G	CF		03	FB	00393	CALLS	#3, PARSE_COMMA_LIST		0730
			1C	DD	00398	PUSHL	#28		
		0170	C6	9F	0039A	PUSHAB	D SPACE		
0000G	CF		02	FB	0039E	CALLS	#2, PARSE_IF_TRUE		0736
			2D	DD	003A3	PUSHL	#45		
			28	DD	003A5	PUSHL	#43		
		0180	C6	9F	003A7	PUSHAB	D TRAILER		
0000G	CF		03	FB	003AB	CALLS	#3, PARSE_LOCAL_TRUE_FALSE		0749
	6E		02	DD	003B0	MOVL	#2, (SP)		
			5E	DD	003B3	PUSHL	SP		0744
			7E	7C	003B5	CLRG	-(SP)		
		F0	AD	9F	003B7	PUSHAB	FIND_DEFAULT		
		14	AE	9F	003BA	PUSHAB	FIND_CONTEXT		
		F8	AD	9F	003BD	PUSHAB	FIND_RESULT		
		E0	AD	9F	003C0	PUSHAB	INPUT_DESC		
00000000G	00		07	FB	003C3	CALLS	#7, LIB\$FIND_FILE		
	55		50	DD	003CA	MOVL	R0, STATUS_2		0750
000182CA	8F		55	D1	003CD	CMPL	STATUS_2, #99018		
			68	13	003D4	BEQL	17\$		
	52	04	AE	DD	003D6	MOVL	FIND_CONTEXT, R2		0762
	75		55	E9	003DA	BLBC	STATUS_2, 19\$		0758
08	AE	F8	AD	9E	003DD	MOVAB	FIND_RESULT, PROMPT_DESC		0769
			7E	D4	003E2	CLRL	-(SP)		0772
		0C	AE	9F	003E4	PUSHAB	PROMPT_DESC		
	07		67	E9	003E7	BLBC	FLAGS, 14\$		0776
	50	02A4	C6	9E	003EA	MOVAB	P.ADW, R0		0777
			05	11	003EF	BRB	15\$		
	50	02C0	C6	9E	003F1	MOVAB	P.ADY, R0		0778
			50	DD	003F6	PUSHL	R0		
			7E	D4	003F8	CLRL	-(SP)		0772
			52	DD	003FA	PUSHL	R2		0774
		F8	A7	9F	003FC	PUSHAB	QUAL_CONTEXT		0772
00000000G	00		06	FB	003FF	CALLS	#6, LIB\$QUAL_FILE_MATCH		
	53		50	DD	00406	MOVL	R0, STATUS_3		0783
	35		53	E8	00409	BLBS	STATUS_3, 18\$		
00000000G	8F		53	D1	0040C	CMPL	STATUS_3, #LIB\$FILFAIMAT		0786
			20	13	00413	BEQL	16\$		
00000000G	8F		53	D1	00415	CMPL	STATUS_3, #LIB\$QUIPRO		0787
			17	13	0041C	BEQL	16\$		
00000000G	8F		53	D1	0041E	CMPL	STATUS_3, #LIB\$NEGANS		0788
			0E	13	00425	BEQL	16\$		
		F8	AD	9F	00427	PUSHAB	FIND_RESULT		0791
			01	DD	0042A	PUSHL	#1		
		00000000*	8F	DD	0042C	PUSHL	#<<SUBMIT\$FACILITY@16>+4920>		0792
			00BD	31	00432	BRW	30\$		
00000000G	8F		53	D1	00435	CMPL	STATUS_3, #LIB\$QUIPRO		0795
			35	12	0043C	BNEQ	22\$		
		FEBD	31	0043E	BRW	10\$			0797
05	A2	60	8F	8A	00441	BICB2	#96, 5(R2)		0802
			52	DD	00446	PUSHL	R2		0804
00000000G	00		01	FB	00448	CALLS	#1, SYS\$OPEN		
	24		50	E8	0044F	BLBS	R0, 23\$		
	7E	08	A2	7D	00452	MOVQ	8(R2), -(SP)		0811

			F8	AD	9F	00456	PUSHAB	FIND_RESULT	0806
				01	DD	00459	PUSHL	#1	
	08			67	E9	0045B	BLBC	FLAGS, 20\$	0807
		00000000*		8F	DD	0045E	PUSHL	#<<<SUBMIT\$ FACILITY@16>+4248>+2>	0808
				06	11	00464	BRB	21\$	
		00000000*		8F	DD	00466	PUSHL	#<<<SUBMIT\$ FACILITY@16>+4248>+4>	0809
	00000000G	00		05	FB	0046C	CALLS	#5, LIB\$SIGNAL	0807
			FF	3A	31	00473	BRW	13\$	0806
27		67		02	E1	00476	BBC	#2, FLAGS, 26\$	0814
14		A2		05	E0	0047A	BBS	#5, 65(R2), 24\$	0817
	41		F8	AD	9F	0047F	PUSHAB	FIND_RESULT	0819
				01	DD	00482	PUSHL	#1	
		00000000G		8F	DD	00484	PUSHL	#SUBMIT\$ REMOINSF	
	00000000G	00		03	FB	0048A	CALLS	#3, LIB\$SIGNAL	
				0E	11	00491	BRB	26\$	
	06			67	E9	00493	BLBC	FLAGS, 25\$	0823
	05	A2		20	88	00496	BISB2	#32, 5(R2)	0824
				05	11	0049A	BRB	26\$	
	05	A2	40	8F	88	0049C	BISB2	#64, 5(R2)	0825
				52	DD	004A1	PUSHL	R2	0829
	00000000G	00		01	FB	004A3	CALLS	#1, SYS\$CLOSE	
		16		50	E8	004AA	BLBS	R0, 27\$	
		7E	08	A2	7D	004AD	MOVQ	8(R2), -(SP)	0834
			F8	AD	9F	004B1	PUSHAB	FIND_RESULT	0831
				01	DD	004B4	PUSHL	#1	
		00000000*		8F	DD	004B6	PUSHL	#<<<SUBMIT\$ FACILITY@16>+4176>+2>	0832
	00000000G	00		05	FB	004BC	CALLS	#5, LIB\$SIGNAL	
AC		67		02	E0	004C3	BBS	#2, FLAGS, 22\$	0837
OD		A2		05	E1	004C7	BBC	#5, 65(R2), 28\$	0840
	41		F8	AD	9F	004CC	PUSHAB	FIND_RESULT	0842
				01	DD	004CF	PUSHL	#1	
		00000000G		8F	DD	004D1	PUSHL	#SUBMIT\$ REMOINSF	
				19	11	004D7	BRB	30\$	
09	41	A2		06	E1	004D9	BBC	#6, 65(R2), 29\$	0847
		05	43	A2	E8	004DE	BLBS	67(R2), 29\$	0848
14	40	A2		05	E1	004E2	BBC	#5, 64(R2), 31\$	0849
			F8	AD	9F	004E7	PUSHAB	FIND_RESULT	0851
				01	DD	004EA	PUSHL	#1	
		00000000G		8F	DD	004EC	PUSHL	#SUBMIT\$ INVFILE\$PE	
	00000000G	00		03	FB	004F2	CALLS	#3, LIB\$SIGNAL	
				4A	11	004F9	BRB	34\$	
	50	28		A2	D0	004FB	MOVL	40(R2), NAM	0864
	6B	0027001C		8F	D0	004FF	MOVL	#2555932, (Q ICURSOR)	0866
	04	AB	14	AD	9E	00506	MOVAB	20(R0), 4(Q ICURSOR)	0868
			08	AB	7C	0050B	CLRQ	8(Q ICURSOR)	0869
				7E	7C	0050E	CLRQ	-(SP)	0878
		14		AE	9F	00510	PUSHAB	IOSB	
		0434		CE	9F	00513	PUSHAB	ITEM_BUFFER	
	7E			02	7D	00517	MOVQ	#2, -(SP)	
				7E	D4	0051A	CLRL	-(SP)	
	00000000G	00		07	FB	0051C	CALLS	#7, SYS\$SNDJBCW	
		07		50	E9	00523	BLBC	STATUS, 32\$	0879
		50	0C	AE	D0	00526	MOVL	IOSB, STATUS	
		16		50	E8	0052A	BLBS	STATUS, 33\$	0880
				50	DD	0052D	PUSHL	STATUS	0884
			F8	AD	9F	0052F	PUSHAB	FIND_RESULT	0882
				01	DD	00532	PUSHL	#1	

00000000G	00	00000000G	8F	DD	00534	PUSHL	#SUBMIT\$ ADDFIL	
			04	FB	0053A	CALLS	#4, LIB\$SIGNAL	
			02	11	00541	BRB	34\$	
			54	D6	00543	INCL	CNT	0886
			FE68	31	00545	BRW	13\$	0837
03	67		02	E1	00548	BBC	#2, FLAGS, 36\$	0896
			0087	31	0054C	BRW	40\$	
	5B	0428	CE	9E	0054F	MOVAB	ITEM_BUFFER, Q_ICURSOR	0906
1C	67		01	E1	00554	BBC	#1, FLAGS, 37\$	0907
		FF4C	CD	D4	00558	CLRL	JSM_DESC	0910
	FF50	FF54	CD	9E	0055C	MOVAB	JSM_BUFFER, JSM_DESC+4	0911
	8B	00580084	8F	D0	00563	MOVL	#5767300, (Q_ICURSOR)+	0912
	8B	FF54	CD	9E	0056A	MOVAB	JSM_BUFFER, (Q_ICURSOR)+	0914
	8B	FF4C	CD	9E	0056F	MOVAB	JSM_DESC, (Q_ICURSOR)+	0915
			6B	D4	00574	CLRL	(Q_ICURSOR)	0918
			7E	7C	00576	CLRQ	-(SP)	0926
		14	AE	9F	00578	PUSHAB	IOSB	
		0434	CE	9F	0057B	PUSHAB	ITEM_BUFFER	
	7E		09	7D	0057F	MOVQ	#9, -(SP)	
			7E	D4	00582	CLRL	-(SP)	
00000000G	00		07	FB	00584	CALLS	#7, SYS\$SNDJBCW	
	53		50	D0	0058B	MOVL	R0, STATUS	
		14	AE	9F	0058E	PUSHAB	EXIT_BLOCK	0931
00000000G	00		01	FB	00591	CALLS	#1, SYS\$SCANEXH	
	07		53	E9	00598	BLBC	STATUS, 38\$	0937
	53		AE	D0	0059B	MOVL	IOSB, STATUS	
	15	0C	53	E8	0059F	BLBS	STATUS, 39\$	0938
			54	D5	005A2	TSTL	CNT	
			11	13	005A4	BEQL	39\$	
			53	DD	005A6	PUSHL	STATUS	0941
			7E	D4	005A8	CLRL	-(SP)	0940
		00000000G	8F	DD	005AA	PUSHL	#SUBMIT\$ CREJOB	
00000000G	00		03	FB	005B0	CALLS	#3, LIB\$STOP	
1B	67		01	E1	005B7	BBC	#1, FLAGS, 40\$	0947
			54	D5	005BB	TSTL	CNT	
			17	13	005BD	BEQL	40\$	
		FF4C	CD	9F	005BF	PUSHAB	JSM_DESC	0949
		00010001	8F	DD	005C3	PUSHL	#65537	0951
		00000000*	8F	DD	005C9	PUSHL	#<<<SUBMIT\$ FACILITY@16>+4400>+1>	0950
00000000G	00		03	FB	005CF	CALLS	#3, LIB\$SIGNAL	
50	FC	A7 10000000	8F	C9	005D6	BISL3	#268435456, EXIT_STATUS, R0	0958
			04	005DF	RET			0959

; Routine Size: 1504 bytes, Routine Base: CODE + 0360

```
0960 1 ROUTINE GET_COMM_QUAL_CONTEXT: NOVALUE =
0961 1 ++
0962 1
0963 1 FUNCTIONAL DESCRIPTION:
0964 1 This routine is gets the common qualifier context.
0965 1
0966 1 INPUT PARAMETERS:
0967 1 NONE
0968 1
0969 1 IMPLICIT INPUTS:
0970 1 QUAL_CONTEXT - defined locally in main routine.
0971 1
0972 1 OUTPUT PARAMETERS:
0973 1 NONE
0974 1
0975 1 IMPLICIT OUTPUTS:
0976 1 NONE
0977 1
0978 1 ROUTINE VALUE:
0979 1 NONE
0980 1
0981 1 SIDE EFFECTS:
0982 1 NONE
0983 1
0984 1 --
0985 1
0986 2 BEGIN
0987 2
0988 2 LOCAL
0989 2 BITMAP,
0990 2 STATUS;
0991 2
0992 2 ! Set up the context bitmap
0993 2
0994 2 BITMAP= LIBSM_CQF_CONFIRM OR LIBSM_CQF_EXCLUDE OR
0995 2 LIBSM_CQF_BEFORE OR LIBSM_CQF_SINCE OR
0996 2 LIBSM_CQF_CREATED OR LIBSM_CQF_MODIFIED OR
0997 2 LIBSM_CQF_EXPIRED OR LIBSM_CQF_BACKUP OR
0998 2 LIBSM_CQF_BYOWNER;
0999 2
1000 2 ! Get the common qualifier context
1001 2
1002 2 IF NOT (STATUS = LIBSQUAL_FILE_PARSE (BITMAP, QUAL_CONTEXT))
1003 2 THEN
1004 2 SIGNAL_STOP (.STATUS);
1005 2
1006 1 END;
```

```
0000 00000 GET_COMM_QUAL_CONTEXT:
7E 01FF 8F 3C 00002 .WORD Save nothing
0000 04 CF 9F 00007 MOVZWL #511, BITMAP
04 AE 9F 0000B PUSHAB QUAL_CONTEXT
PUSHAB BITMAP
```

```
0960
0997
1002
```

PRINT and SUBMIT Commands

M 12
16-Sep-1984 00:28:51 YAX-11 B11ss-32 V4.0-742
14-Sep-1984 12:09:55 [CLIUTL.SRC]SUBMIT.832;1

Page 30
(4)

00000000G 00
09

```

02 FB 0000E
50 EB 00015
50 DD 00018
01 FB 0001A
    04 00021 18:

```

```
CALLS      #2, LIB$QUAL_FILE_PARSE
BLBS       STATUS, 13
PUSHL      STATUS
CALLS      #1, LIB$STOP
RET
```

1004
1006

000000000G 00

; Routine Size: 34 bytes, Routine Base: CODE + 0940


```
902 1007 1 ROUTINE EXIT_HANDLER: NOVALUE=
903 1008 1
904 1009 1 ++
905 1010 1
906 1011 1 FUNCTIONAL DESCRIPTION:
907 1012 1 This routine is an exit handler that is established as long as the
908 1013 1 process owns an open job. Its function is to delete the open job
909 1014 1 if an image exit should occur.
910 1015 1
911 1016 1 INPUT PARAMETERS:
912 1017 1 Standard exit handler parameters (not used).
913 1018 1
914 1019 1 IMPLICIT INPUTS:
915 1020 1 NONE
916 1021 1
917 1022 1 OUTPUT PARAMETERS:
918 1023 1 NONE
919 1024 1
920 1025 1 IMPLICIT OUTPUTS:
921 1026 1 NONE
922 1027 1
923 1028 1 ROUTINE VALUE:
924 1029 1 NONE
925 1030 1
926 1031 1 SIDE EFFECTS:
927 1032 1 NONE
928 1033 1
929 1034 1 --
930 1035 1
931 1036 2 BEGIN
932 1037 2 $SNDJBC(FUNC=SJC$_CLOSE_DELETE);
933 1038 1 END;
```

```
                                .EXTRN  SYS$$SNDJBC
                                0000 00000 EXIT_HANDLER:
                                .WORD    Save nothing
                                7E 7C 00002 CLRQ    -(SP)
                                7E 7C 00004 CLRQ    -(SP)
                                08 7D 00006 MOVQ    #8, -(SP)
                                7E D4 00009 CLRL    -(SP)
                                00000000G 00 07 FB 0000B CALLS  #7, SYS$$SNDJBC
                                04 00012 RET
```

; Routine Size: 19 bytes, Routine Base: CODE + 0962

1007
1037
1038

```

935 1039 1 ROUTINE HANDLER(SIG,MECH)=
936 1040 1
937 1041 1 ++
938 1042 1
939 1043 1 FUNCTIONAL DESCRIPTION:
940 1044 1 This routine is the condition handler for the main routine. It
941 1045 1 saves the most severe condition as the exit status.
942 1046 1
943 1047 1 INPUT PARAMETERS:
944 1048 1 Standard VMS condition handler parameters.
945 1049 1
946 1050 1 IMPLICIT INPUTS:
947 1051 1 NONE
948 1052 1
949 1053 1 OUTPUT PARAMETERS:
950 1054 1 NONE
951 1055 1
952 1056 1 IMPLICIT OUTPUTS:
953 1057 1 EXIT_STATUS - Contains the most severe status encountered.
954 1058 1
955 1059 1 ROUTINE VALUE:
956 1060 1 SSS_RESIGNAL
957 1061 1
958 1062 1 SIDE EFFECTS:
959 1063 1 NONE
960 1064 1
961 1065 1 --
962 1066 1
963 1067 2 BEGIN
964 1068 2 MAP
965 1069 2 SIG: REF BBLOCK, ! Signal parameters
966 1070 2 MECH: REF BBLOCK; ! Mechanism parameters
967 1071 2 BIND
968 1072 2 SIGNAME = SIG[CHF$$_SIG_NAME] : BBLOCK; ! Name of signal
969 1073 2
970 1074 2
971 1075 2 ! Change the facility code from SUBMIT to PRINT if required.
972 1076 2
973 1077 2 IF
974 1078 2 .FLAGS[V_PRINT]
975 1079 2 AND NOT .SIGNAME
976 1080 2 AND .SIGNAME[STSSV_FAC_NO] EQL SUBMIT$_FACILITY
977 1081 2 THEN
978 1082 2 SIGNAME[STSSV_FAC_NO] = PRINT$_FACILITY;
979 1083 2
980 1084 2
981 1085 2 ! Save the most severe condition as the exit status.
982 1086 2
983 1087 2 IF
984 1088 2 NOT .SIGNAME ! If an error signal
985 1089 2 AND ((.SIGNAME[STSSV_SEVERITY] ! and severity is worse
986 1090 2 GTRU .EXIT_STATUS[STSSV_SEVERITY])
987 1091 2 OR .EXIT_STATUS[STSSV_SEVERITY]) ! or no errors yet
988 1092 2 THEN
989 1093 2 EXIT_STATUS = .SIGNAME; ! then save it for exit
990 1094 2
991 1095 2
```

SUBMIT
V04-000

PRINT and SUBMIT Commands

C 13
16-Sep-1984 00:28:51
14-Sep-1984 12:09:55

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SUBMIT.B32;1

Page 33
(6)

: 992
: 993
1096 2 \$\$\$ RESIGNAL
1097 1 END;

! Resignal to get message

00000000G	BF	02	A0	50	04	52	0000'	CF	9E	00002	HANDLER: .WORD	Save R2	: 1039
						AC		04	C1	00007	MOVAB	EXIT_STATUS, R2	: 1072
						19	04	A2	E9	0000C	ADDL3	#4, SIG, R0	: 1078
						2B		60	E8	00010	BLBC	FLAGS, 1\$: 1079
						0C		00	ED	00013	BLBS	(R0), 3\$: 1080
								0A	12	0001D	CMPZV	#0, #12, 2(R0), #SUBMITS_FACILITY	
								8F	F0	0001F	BNEQ	1\$	
02	A0		0C			12	00000000G	60	E8	00029	INSV	#PRINTS_FACILITY, #0, #12, 2(R0)	: 1082
						03		00	EF	0002C	BLBS	(R0), 3\$: 1088
	51		62			03		00	ED	00031	EXTZV	#0, #3, EXIT_STATUS, R1	: 1090
	51		60			03		03	1A	00036	CMPZV	#0, #3, (R0), R1	
						03		62	E9	00038	BGTRU	2\$	
						62		60	D0	0003B	BLBC	EXIT_STATUS, 3\$: 1091
						50	0918	8F	3C	0003E	MOVL	(R0), EXIT_STATUS	: 1093
								04	00043		MOVZWL	#2328, R0	: 1097
											RET		

; Routine Size: 68 bytes, Routine Base: CODE + 0975

SUBMIT
V04-000

PRINT and SUBMIT Commands

D 13
16-Sep-1984 00:28:51
14-Sep-1984 12:09:55

VAX-11 Bliss-32 V4.0-742
[CLIUTL.SRC]SUBMIT.B32;1

Page 34
(7)

: 995
: 996
1098 1 END
1099 0 ELUDOM

.EXTRN LIB\$SIGNAL, LIB\$STOP

PSECT SUMMARY

Name	Bytes	Attributes
DATA	12	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
CODE	2489	NOVEC, NOWRT, RD, EXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
_\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	107	1	581	00:00.9

COMMAND QUALIFIERS

: BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:SUBMIT/OBJ=OBJ\$:SUBMIT MSRC\$:SUBMIT/UPDATE=(ENH\$:SUBMIT)

: Size: 1625 code + 876 data bytes
: Run Time: 00:32.4
: Elapsed Time: 01:47.3
: Lines/CPU Min: 2037
: Lexemes/CPU-Min: 19676
: Memory Used: 440 pages
: Compilation Complete

0059 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200
201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400
401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500
501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	524	525	526	527	528	529	530	531	532	533	534	535	536	537	538	539	540	541	542	543	544	545	546	547	548	549	550	551	552	553	554	555	556	557	558	559	560	561	562	563	564	565	566	567	568	569	570	571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600
601	602	603	604	605	606	607	608	609	610	611	612	613	614	615	616	617	618	619	620	621	622	623	624	625	626	627	628	629	630	631	632	633	634	635	636	637	638	639	640	641	642	643	644	645	646	647	648	649	650	651	652	653	654	655	656	657	658	659	660	661	662	663	664	665	666	667	668	669	670	671	672	673	674	675	676	677	678	679	680	681	682	683	684	685	686	687	688	689	690	691	692	693	694	695	696	697	698	699	700
701	702	703	704	705	706	707	708	709	710	711	712	713	714	715	716	717	718	719	720	721	722	723	724	725	726	727	728	729	730	731	732	733	734	735	736	737	738	739	740	741	742	743	744	745	746	747	748	749	750	751	752	753	754	755	756	757	758	759	760	761	762	763	764	765	766	767	768	769	770	771	772	773	774	775	776	777	778	779	780	781	782	783	784	785	786	787	788	789	790	791	792	793	794	795	796	797	798	799	800
801	802	803	804	805	806	807	808	809	810	811	812	813	814	815	816	817	818	819	820	821	822	823	824	825	826	827	828	829	830	831	832	833	834	835	836	837	838	839	840	841	842	843	844	845	846	847	848	849	850	851	852	853	854	855	856	857	858	859	860	861	862	863	864	865	866	867	868	869	870	871	872	873	874	875	876	877	878	879	880	881	882	883	884	885	886	887	888	889	890	891	892	893	894	895	896	897	898	899	900
901	902	903	904	905	906	907	908	909	910	911	912	913	914	915	916	917	918	919	920	921	922	923	924	925	926	927	928	929	930	931	932	933	934	935	936	937	938	939	940	941	942	943	944	945	946	947	948	949	950	951	952	953	954	955	956	957	958	959	960	961	962	963	964	965	966	967	968	969	970	971	972	973	974	975	976	977	978	979	980	981	982	983	984	985	986	987	988	989	990	991	992	993	994	995	996	997	998	999	1000